IN THE CLAIMS

Please amend claims 1, 10 and 19 as follows:

1. (CURRENTLY AMENDED) A method of optimizing a query in a computer system, the query being performed by the computer system to retrieve data from a database stored on the computer system, the method comprising:

(a) identifying a correlated subquery in the query that qualifies for transformation;

(b) transitively closing a local filtering predicate within the identified correlated subquery that is on a same column as a correlation join predicate within the identified correlated subquery; [[and]]

(c) pulling out the transitively closed predicate from the correlated subquery to a parent query block of the query in order to create an optimized query; and

(d) performing the optimized query in the computer system to retrieve data from the database.

2. (ORIGINAL) The method of claim 1, wherein the identifying step comprises recognizing that the correlated subquery is a Boolean term predicate that does not contain negation and the correlated subquery does not contain a COUNT arithmetic function.

3. (ORIGINAL) The method of claim 2, wherein the correlated subquery comprises an EXISTS correlated subquery, COL IN correlated subquery, COL = ANY correlated subquery, or COL = correlated subquery, but not a NOT EXISTS correlated subquery or a correlated subquery that contains a COUNT arithmetic function.

4. (ORIGINAL) The method of claim 2, wherein the identifying step comprises recognizing that the correlated subquery includes the local filtering predicate on the same column as the correlation join predicate.

5. (ORIGINAL) The method of claim 1, wherein the transitively closing step comprises performing a predicate transitive closure of the local filtering predicate.

6. (ORIGINAL) The method of claim 1, wherein the pulling out step comprises bubbling up the transitively closed predicate to the parent query block of the query.

7. (ORIGINAL) The method of claim 1, wherein the transitively closed predicate is applied to a table referenced in the parent query block.

8. (ORIGINAL) The method of claim 7, further comprising accessing the table referenced by the transitively closed predicate via a new access path.

9. (ORIGINAL) The method of claim 6 or 7, further comprising applying the transitively closed predicate prior to performing the correlated subquery, thereby resulting in fewer executions of the correlated subquery.

10. (CURRENTLY AMENDED) A computer-implemented apparatus for optimizing a query, comprising:

a computer system;

logic, performed by the computer system, for

(a) identifying a correlated subquery in the query that qualifies for transformation;

(b) transitively closing a local filtering predicate within the identified correlated subquery that is on a same column as a correlation join predicate within the identified correlated subquery; [[and]]

(c) pulling out the transitively closed predicate from the correlated subquery to a parent query block of the query in order to create an optimized query; and

(d) performing the optimized query in the computer system to retrieve data from the database.

11. (ORIGINAL) The apparatus of claim 10, wherein the logic for identifying comprises logic for recognizing that the correlated subquery is a Boolean term predicate that does not contain negation and the correlated subquery does not contain a COUNT arithmetic function.

12. (ORIGINAL) The apparatus of claim 11, wherein the correlated subquery comprises an EXISTS correlated subquery, COL IN correlated subquery, COL = ANY correlated subquery, or COL = correlated subquery, but not a NOT EXISTS correlated subquery or a correlated subquery that contains a COUNT arithmetic function.

13. (ORIGINAL) The apparatus of claim 11, wherein the logic for identifying comprises logic for recognizing that the correlated subquery includes the local filtering predicate on the same column as the correlation join predicate.

14. (ORIGINAL) The apparatus of claim 10, wherein the logic for transitively closing comprises logic for performing a predicate transitive closure of the local filtering predicate.

15. (ORIGINAL) The apparatus of claim 10, wherein the logic for pulling out comprises logic for bubbling up the transitively closed predicate to the parent query block of the query.

16. (ORIGINAL) The apparatus of claim 10, wherein the transitively closed predicate is applied to a table referenced in the parent query block.

17. (ORIGINAL) The apparatus of claim 16, further comprising logic for accessing the table referenced by the transitively closed predicate via a new access path.

18. (ORIGINAL) The apparatus of claim 15 or 16, further comprising logic for applying the transitively closed predicate prior to performing the correlated subquery, thereby resulting in fewer executions of the correlated subquery.

19. (CURRENTLY AMENDED) An article of manufacture comprising a computer-readable storage device embodying ~~logic for~~ instructions that, when read and executed by a computer system, results in the computer system performing a method for optimizing a query, the query being performed by [[a]] the computer system to retrieve data from a database stored in a data storage device coupled to the computer system, the method comprising:

(a) identifying a correlated subquery in the query that qualifies for transformation;

(b) transitively closing a local filtering predicate within the identified correlated subquery that is on a same column as a correlation join predicate within the identified correlated subquery; [[and]]

(c) pulling out the transitively closed predicate from the correlated subquery to a parent query block of the query in order to create an optimized query; and

(d) performing the optimized query in the computer system to retrieve data from the database.

20. (ORIGINAL) The article of claim 19, wherein the identifying step comprises recognizing that the correlated subquery is a Boolean term predicate that does not contain negation and the correlated subquery does not contain a COUNT arithmetic function.

21. (ORIGINAL) The article of claim 20, wherein the correlated subquery comprises an EXISTS correlated subquery, COL IN correlated subquery, COL = ANY correlated subquery, or COL = correlated subquery, but not a NOT EXISTS correlated subquery or a correlated subquery that contains a COUNT arithmetic function.

22. (ORIGINAL) The article of claim 20, wherein the identifying step comprises recognizing that the correlated subquery includes the local filtering predicate on the same column as the correlation join predicate.

23. (ORIGINAL) The article of claim 19, wherein the transitively closing step comprises performing a predicate transitive closure of the local filtering predicate.

24. (ORIGINAL) The article of claim 19, wherein the pulling out step comprises bubbling up the transitively closed predicate to the parent query block of the query.

25. (ORIGINAL) The article of claim 19, wherein the transitively closed predicate is applied to a table referenced in the parent query block.

26. (ORIGINAL) The article of claim 25, further comprising accessing the table referenced by the transitively closed predicate via a new access path.

27. (ORIGINAL) The article of claim 24 or 25, further comprising applying the transitively closed predicate prior to performing the correlated subquery, thereby resulting in fewer executions of the correlated subquery.